



```
PPPPPPPP  LL      IIIIII  GGGGGGGG  EEEEEEEEE  TTTTTTTTT  EEEEEEEEE  DDDDDDDD  IIIIII
PPPPPPPP  LL      IIIIII  GGGGGGGG  EEEEEEEEE  TTTTTTTTT  EEEEEEEEE  DDDDDDDD  IIIIII
PP      PP  LL      II      GG      EE      TT      EE      DD      DD  II
PP      PP  LL      II      GG      EE      TT      EE      DD      DD  II
PP      PP  LL      II      GG      EE      TT      EE      DD      DD  II
PP      PP  LL      II      GG      EE      TT      EE      DD      DD  II
PPPPPPPP  LL      II      GG      EEEEEEE  TT      EEEEEEE  DD      DD  II
PPPPPPPP  LL      II      GG      EEEEEEE  TT      EEEEEEE  DD      DD  II
PP      LL      II      GG      EE      TT      EE      DD      DD  II
PP      LL      II      GG      EE      TT      EE      DD      DD  II
PP      LL      II      GG      EE      TT      EE      DD      DD  II
PP      LL      II      GG      EE      TT      EE      DD      DD  II
PP      LL      II      GG      EE      TT      EE      DD      DD  II
PP      LLLLLLLLLL  IIIIII  GGGGGG  EEEEEEEEE  TT      EEEEEEEEE  DDDDDDDD  IIIIII
PP      LLLLLLLLLL  IIIIII  GGGGGG  EEEEEEEEE  TT      EEEEEEEEE  DDDDDDDD  IIIIII
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
```



```
0000 1      .title pli$geteitem
0000 2      .ident /1-002/
0000 3
0000 4      ;
0000 5      ;*****
0000 6      ;
0000 7      ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      ;*  ALL RIGHTS RESERVED.
0000 10     ;*
0000 11     ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     ;*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     ;*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     ;*  TRANSFERRED.
0000 17     ;*
0000 18     ;*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     ;*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     ;*  CORPORATION.
0000 21     ;*
0000 22     ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     ;*
0000 25     ;*
0000 26     ;*****
0000 27     ;
0000 28     ;
0000 29     ;
0000 30     ;++
0000 31     ; facility:
0000 32     ;
0000 33     ;     VAX/VMS PL1 runtime library
0000 34     ;
0000 35     ; abstract:
0000 36     ;
0000 37     ;     This module contains the pl1 runtime routines to get items from
0000 38     ;     pl1 stream file under edit controlled format.
0000 39     ;
0000 40     ;
0000 41     ; author: c. spitz 28-nov-79
0000 42     ;
0000 43     ; modified:
0000 44     ;
0000 45     ;
0000 46     ;     1-002  Bill Matthews  29-September-1982
0000 47     ;
0000 48     ;     Invoke macros $defdat and rtshare instead of $defopr and share.
0000 49     ;
0000 50     ; --
0000 51     ;
0000 52     ;
0000 53     ; external definitions
0000 54     ;
0000 55     ;     $deffcb      ;define file control block
0000 56     ;     $defstk     ;define stack frame offsets
0000 57     ;     $defstr     ;define stream block offsets
```



```
0000 58      $defdat      ;define operand node data types
0000 59      $defgetopt    ;define get options block
0000 60      $defcvtind    ;define convert indices
0000 61      $rabdef       ;define rms rab offsets
0000 62      $rmsdef       ;define rms error codes
0000 63
0000 64      ;
0000 65      ; local data
0000 66      ;
0000 67
0000 68      rtshare          ;sharable
0000 69
0000 70
0000 71      ;++
0000 72      ; pli$gete****
0000 73      ;
0000 74      ; the pli$gete**** routines are called by the compiled code to get items
0000 75      ; from a stream input file under edit directed transmission. each routine
0000 76      ; saves the target item descriptor, calls pli$$getfmt_r6 to get the
0000 77      ; next field based on the format, and then calls to pli$rt_cvrt_r3 to store
0000 78      ; the item in the target.
0000 79      ;--
0000 80
0000 81      ;pli$getechar_r6
0000 82      ; inputs:
0000 83      ;     r0 - address of element to get
0000 84      ;     r1 - size/prec of element to get
0000 85      ;     r11 - address of stream block
0000 86      ;     ap - address of file control block
0000 87      ; outputs:
0000 88      ;     none
0000 89      ; side effects:
0000 90      ;     r0-r6 are destroyed
0000 91
0000 92      pli$getechar_r6::
0000 93      bisl    #atr_m_recur,fcbl_attr(ap) ;set recursion flag
0000 94      bicl    #atr_m_flttrg,fcbl_attr(ap) ;clr floating target flag
0000 95      movq    r0,-(sp) ;save destination
0000 96      jsb     g^pli$$getfmt_r6 ;get format
0000 97      movq    (sp)+,r2 ;restore destination
0000 98      addl    #cvt_k_dst_char,r4 ;update case index for char dest
0000 99      calls   #0,g^pli$cvrt_cg_r3 ;convert it
0000 100     bicl    #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
0000 101     rsb     ;return
0000 102
0000 103     ;pli$getevcha_r6
0000 104     ; inputs:
0000 105     ;     r0 - address of element to get
0000 106     ;     r1 - size/prec of element to get
0000 107     ;     r11 - address of stream block
0000 108     ;     ap - address of file control block
0000 109     ; outputs:
0000 110     ;     none
0000 111     ; side effects:
0000 112     ;     r0-r6 are destroyed
0000 113     ;
0000 114
```

|       |          |    |    |      |     |
|-------|----------|----|----|------|-----|
| OC AC | 0C AC    | 08 | C8 | 0000 | 93  |
|       | 04000000 | 8F | CA | 0004 | 94  |
|       | 7E       | 50 | 7D | 000C | 95  |
|       | 00000000 | GF | 16 | 000F | 96  |
|       | 52       | 8E | 7D | 0015 | 97  |
|       | 54       | 05 | C0 | 0018 | 98  |
|       | 00000000 | GF | FB | 001B | 99  |
|       | 0C AC    | 08 | CA | 0022 | 100 |
|       |          |    | 05 | 0026 | 101 |
|       |          |    |    | 0027 | 102 |
|       |          |    |    | 0027 | 103 |
|       |          |    |    | 0027 | 104 |
|       |          |    |    | 0027 | 105 |
|       |          |    |    | 0027 | 106 |
|       |          |    |    | 0027 | 107 |
|       |          |    |    | 0027 | 108 |
|       |          |    |    | 0027 | 109 |
|       |          |    |    | 0027 | 110 |
|       |          |    |    | 0027 | 111 |
|       |          |    |    | 0027 | 112 |
|       |          |    |    | 0027 | 113 |
|       |          |    |    | 0027 | 114 |



```
0027 115 pli$getevcha_r6::
OC AC 0C AC 08 C8 0027 116      bisl      #atr_m_recur,fcbl_attr(ap) ;set recursion flag
      04000000 8F CA 002B 117      bicl      #atr_m_flgtrg,fcbl_attr(ap) ;clr floating target flag
      7E 50 7D 0033 118      clrw      (r0) ;clear length
      00000000 GF 16 0035 119      movq      r0,-(sp) ;save destination
      52 8E 7D 0038 120      jsb      g^pli$$getfmt_r6 ;get format
      54 06 C0 003E 121      movq      (sp)+,r2 ;restore destination
00000000 GF 00 FB 0041 122      addl      #cvt_k_dst_vcha,r4 ;update case index for vcha dest
      OC AC 08 CA 0044 123      calls     #0,g^pli$cvrt_cg_r3 ;convert it
      05 004F 124      bicl      #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
      0050 125      rsb ;return
      0050 126
      0050 127 ;pli$getebit_r6
      0050 128      inputs:
      0050 129      r0 - address of element to get
      0050 130      r1 - size/prec of element to get
      0050 131      r2 - offset of element to get
      0050 132      r11 - address of stream block
      0050 133      ap - address of file control block
      0050 134      outputs:
      0050 135      none
      0050 136      side effects:
      0050 137      r0-r6 are destroyed
      0050 138
      0050 139
      0050 140 pli$getebit_r6::
OC AC 0C AC 08 C8 0050 141      bisl      #atr_m_recur,fcbl_attr(ap) ;set recursion flag
      04000000 8F CA 0054 142      bicl      #atr_m_flgtrg,fcbl_attr(ap) ;clr floating target flag
      7E 50 7D 005C 143      movq      r0,-(sp) ;save dest
      00000000 GF 16 DD 005F 144      pushl     r2 ;save offset
      52 8E 7D 0061 145      jsb      g^pli$$getfmt_r6 ;get format
      54 07 C0 006A 146      popl      r6 ;restore offset
00000000 GF 00 FB 006D 147      movq      (sp)+,r2 ;restore dest
      OC AC 08 CA 0070 148      addl      #cvt_k_dst_bit,r4 ;update case index for bit dest
      05 0077 149      calls     #0,g^pli$cvrt_cg_r3 ;convert it
      007B 150      bicl      #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
      007C 151      rsb ;return
      007C 152
      007C 153 ;pli$geteabit_r6
      007C 154      inputs:
      007C 155      r0 - address of element to get
      007C 156      r1 - size/prec of element to get
      007C 157      r11 - address of stream block
      007C 158      ap - address of file control block
      007C 159      outputs:
      007C 160      none
      007C 161      side effects:
      007C 162      r0-r6 are destroyed
      007C 163
      007C 164
      007C 165 pli$geteabit_r6::
OC AC 0C AC 08 C8 007C 166      bisl      #atr_m_recur,fcbl_attr(ap) ;set recursion flag
      04000000 8F CA 0080 167      bicl      #atr_m_flgtrg,fcbl_attr(ap) ;clr floating target flag
      7E 50 7D 0088 168      movq      r0,-(sp) ;save destination
      00000000 GF 16 008B 169      jsb      g^pli$$getfmt_r6 ;get format
      52 8E 7D 0091 170      movq      (sp)+,r2 ;restore destination
      54 08 C0 009A 171      addl      #cvt_k_dst_abit,r4 ;update case index for abit dest
```



```
00000000'GF 00 FB 0097 172      calls #0,g^pli$cvrt_cg_r3      ;convert it
OC AC 08 CA 009E 173      bicl #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
05 00A2 174      rsb                                     ;return
00A3 175
00A3 176 ;pli$getefixb_r6
00A3 177 : inputs:
00A3 178 : r0 - address of element to get
00A3 179 : r1 - size/prec of element to get
00A3 180 : r11 - address of stream block
00A3 181 : ap - address of file control block
00A3 182 : outputs:
00A3 183 : none
00A3 184 : side effects:
00A3 185 : r0-r6 are destroyed
00A3 186 :
00A3 187
00A3 188 pli$getefixb_r6::
OC AC 0C AC 08 C8 00A3 189      bisl #atr_m_recur,fcbl_attr(ap) ;set recursion flag
00A7 190      bicl #atr_m_flgtrg,fcbl_attr(ap) ;clr floating target flag
7E 50 7D 00AF 191      movq r0,-(sp) ;save destination
00000000'GF 16 00B2 192      jsb g^pli$getfmt_r6 ;get format
52 8E 7D 00B8 193      movq (sp)+,r2 ;restore destination
54 01 C0 00BB 194      addl #cvt_k_dst_fixb,r4 ;update case index for fixb dest
00000000'GF 00 FB 00BE 195      calls #0,g^pli$cvrt_cg_r3 ;convert it
OC AC 08 CA 00C5 196      bicl #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
05 00C9 197      rsb                                     ;return
00CA 198
00CA 199 ;pli$getefixd_r6
00CA 200 : inputs:
00CA 201 : r0 - address of element to get
00CA 202 : r1 - size/prec of element to get
00CA 203 : r11 - address of stream block
00CA 204 : ap - address of file control block
00CA 205 : outputs:
00CA 206 : none
00CA 207 : side effects:
00CA 208 : r0-r6 are destroyed
00CA 209 :
00CA 210
00CA 211 pli$getefixd_r6::
OC AC 0C AC 08 C8 00CA 212      bisl #atr_m_recur,fcbl_attr(ap) ;set recursion flag
00CE 213      bicl #atr_m_flgtrg,fcbl_attr(ap) ;clr floating target flag
7E 50 7D 00D6 214      movq r0,-(sp) ;save destination
00000000'GF 16 00D9 215      jsb g^pli$getfmt_r6 ;get format
52 8E 7D 00DF 216      movq (sp)+,r2 ;restore destination
54 03 C0 00E2 217      addl #cvt_k_dst_fixd,r4 ;update case index for fixd dest
00000000'GF 00 FB 00E5 218      calls #0,g^pli$cvrt_cg_r3 ;convert it
OC AC 08 CA 00EC 219      bicl #atr_m_recur,fcbl_attr(ap) ;clr recursion flag
05 00F0 220      rsb                                     ;return
00F1 221
00F1 222 ;pli$getefltb_r6
00F1 223 : inputs:
00F1 224 : r0 - address of element to get
00F1 225 : r1 - size/prec of element to get
00F1 226 : r11 - address of stream block
00F1 227 : ap - address of file control block
00F1 228 : outputs:
```



```
00F1 229 : none
00F1 230 : side effects:
00F1 231 : r0-r6 are destroyed
00F1 232 :
00F1 233 :
00F1 234 pli$getefltb_r6::
04000008 8F C8 00F1 235 bisl #<atr_m_recur!atr_m_flttrg>,- :set recursion
OC AC 00F7 236 fcb_l_aatr(ap) :and floating target
08 51 07 E1 00F9 237 bbc #7,r1-10$ :if g float
0000008F 8F DD 00FD 238 pushl #<15+128> :set max g float dec prec
OB 11 0103 239 brb 30$ :cont
35 51 D1 0105 240 10$: cmpl r1,#53 :is it huge?
04 15 0108 241 bleq 20$ :if leq, no, cont
22 DD 010A 242 pushl #34 :set max huge dec prec
02 11 010C 243 brb 30$ :cont
OF DD 010E 244 20$: pushl #15 :set max d float dec prec
0110 245 : (no need to bother w/ f/d since
0110 246 : convert routines go to d)
00000000 7E 50 7D 0110 247 30$: movq r0,-(sp) :save destination
GF 16 0113 248 jsb g^pli$getfmt_r6 :get format
52 8E 7D 0119 249 movq (sp)+,r2 :restore destination
5E 04 AE 9E 011C 250 movab 4(sp),sp :clean dec prec from stack
00000000 54 02 C0 0120 251 addl #cvt_k_dst_fltb,r4 :update case index for fltb dest
GF 00 FB 0123 252 calls #0,g^pli$cvrt_cg_r3 :convert it
04000008 8F CA 012A 253 bicl #<atr_m_recur!atr_m_flttrg>,- :clr recursion flag
OC AC 0130 254 fcb_l_aatr(ap) :and floating source
05 0132 255 rsb :return
0133 256
0133 257 pli$getefltd_r6
0133 258 inputs:
0133 259 r0 - address of element to get
0133 260 r1 - size/prec of element to get
0133 261 r11 - address of stream block
0133 262 ap - address of file control block
0133 263 outputs:
0133 264 none
0133 265 side effects:
0133 266 r0-r6 are destroyed
0133 267
0133 268
0133 269 pli$getefltd_r6::
04000008 8F C8 0133 270 bisl #<atr_m_recur!atr_m_flttrg>,- :set recursion
OC AC 0139 271 fcb_l_aatr(ap) :and floating target
51 DD 013B 272 pushl r1 :set float dec prec
7E 50 7D 013D 273 movq r0,-(sp) :save destination
00000000 GF 16 0140 274 jsb g^pli$getfmt_r6 :get format
52 8E 7D 0146 275 movq (sp)+,r2 :restore destination
5E 04 AE 9E 0149 276 movab 4(sp),sp :clean dec prec from stack
00000000 54 04 C0 014D 277 addl #cvt_k_dst_fltd,r4 :update case index for fltd dest
GF 00 FB 0150 278 calls #0,g^pli$cvrt_cg_r3 :convert it
04000008 8F CA 0157 279 bicl #<atr_m_recur!atr_m_flttrg>,- :clr recursion flag
OC AC 015D 280 fcb_l_aatr(ap) :and floating source
05 015F 281 rsb :return
0160 282
0160 283 pli$getepic_r6
0160 284 inputs:
0160 285 r0 - address of element to get
```

```
0160 286 :      r1 - size/prec of element to get
0160 287 :      r11 - address of stream block
0160 288 :      ap - address of file control block
0160 289 :      outputs:
0160 290 :      none
0160 291 :      side effects:
0160 292 :      r0-r6 are destroyed
0160 293 :
0160 294 pli$getepic r6::
0160 295     bisl    #atr_m_recur, fcb_l_attr(ap) ;set recursion flag
OC AC 04000000 8F CA 0164 296     bicl    #atr_m_flgtrg, fcb_l_attr(ap) ;clr floating target flag
      7E 50 7D 016C 297     movq    r0, -(sp) ;save destination
      00000000 GF 16 016F 298     jsb     g^pli$getfmt_r6 ;get format
      52 8E 7D 0175 299     movq    (sp)+, r2 ;restore destination
      54 00 CO 0178 300     addl    #cvt_k_dst_pic, r4 ;update case index for pic dest
00000000 GF 00 FB 017B 301     calls   #0, g^pli$cvrt_cg_r3 ;convert it
      OC AC 08 CA 0182 302     bicl    #atr_m_recur, fcb_l_attr(ap) ;clr recursion flag
      05 0186 303     rsb         ;return
      0187 304
      0187 305     .end
```



PLISGETEITEM  
Symbol table

L 4

16-SEP-1984 02:19:30 VAX/VMS Macro V04-00  
6-SEP-1984 11:38:17 [PLIRTL.SRC]PLIGETEDI.MAR;1

Page 7  
(1)

|                 |          |          |   |    |
|-----------------|----------|----------|---|----|
| ATR_M_FLTTRG    | =        | 04000000 |   |    |
| ATR_M_RECUR     | =        | 00000008 |   |    |
| CVT_K_DST_ABIT  | =        | 00000008 |   |    |
| CVT_K_DST_BIT   | =        | 00000007 |   |    |
| CVT_K_DST_CHAR  | =        | 00000005 |   |    |
| CVT_K_DST_FIXB  | =        | 00000001 |   |    |
| CVT_K_DST_FIXD  | =        | 00000003 |   |    |
| CVT_K_DST_FLTB  | =        | 00000002 |   |    |
| CVT_K_DST_FLTD  | =        | 00000004 |   |    |
| CVT_K_DST_PIC   | =        | 00000000 |   |    |
| CVT_K_DST_VCHA  | =        | 00000006 |   |    |
| FCB_B_ENVIR     |          | 000001C2 |   |    |
| FCB_B_ESA       |          | 0000012E |   |    |
| FCB_B_EXTRA     |          | 0000003D |   |    |
| FCB_B_FAB       |          | 000000A6 |   |    |
| FCB_B_IDENT     |          | 00000040 |   |    |
| FCB_B_IDENT_NAM |          | 00000042 |   |    |
| FCB_B_NAM       |          | 000000F6 |   |    |
| FCB_B_NUMKCBS   |          | 0000003C |   |    |
| FCB_B_RAB       |          | 00000062 |   |    |
| FCB_C_LEN       |          | 000001C2 |   |    |
| FCB_C_STRLN     |          | 00000034 |   |    |
| FCB_L_ATTR      |          | 0000000C |   |    |
| FCB_L_BUF       |          | 00000014 |   |    |
| FCB_L_BUF_END   |          | 00000018 |   |    |
| FCB_L_BUF_PT    |          | 0000001C |   |    |
| FCB_L_CNDADDR   |          | 000001B2 |   |    |
| FCB_L_CONDIT    |          | 000001AE |   |    |
| FCB_L_DTTR      |          | 00000010 |   |    |
| FCB_L_ERROR     |          | 00000008 |   |    |
| FCB_L_KCB       |          | 00000038 |   |    |
| FCB_L_NEXT      |          | 00000000 |   |    |
| FCB_L_PREVIOUS  |          | 00000004 |   |    |
| FCB_L_PRN       |          | 00000034 |   |    |
| FCB_Q_RFA       |          | 00000020 |   |    |
| FCB_W_COLUMN    |          | 0000002E |   |    |
| FCB_W_IDENT_LEN |          | 00000040 |   |    |
| FCB_W_LINE      |          | 00000030 |   |    |
| FCB_W_LINESIZE  |          | 0000002A |   |    |
| FCB_W_PAGE      |          | 00000032 |   |    |
| FCB_W_PAGESIZE  |          | 0000002C |   |    |
| FCB_W_REVISION  |          | 00000028 |   |    |
| GETOPT_B_BITS   |          | 00000009 |   |    |
| GETOPT_B_TMO    |          | 00000008 |   |    |
| GETOPT_C_LEN    |          | 0000000A |   |    |
| GETOPT_L_FXDCTL |          | 00000000 |   |    |
| GETOPT_L_PROMPT |          | 00000004 |   |    |
| PLISSGETFMT_R6  | *****    |          | X | 02 |
| PLISCVRT CG-R3  | *****    |          | X | 02 |
| PLISGETEABIT_R6 | 0000007C | RG       |   | 02 |
| PLISGETEBIT_R6  | 00000050 | RG       |   | 02 |
| PLISGETECHAR_R6 | 00000000 | RG       |   | 02 |
| PLISGETEFIXB_R6 | 000000A3 | RG       |   | 02 |
| PLISGETEFIXD_R6 | 000000CA | RG       |   | 02 |
| PLISGETEFLTB_R6 | 000000F1 | RG       |   | 02 |
| PLISGETEFLTD_R6 | 00000133 | RG       |   | 02 |
| PLISGETEPIC_R6  | 00000160 | RG       |   | 02 |

|                 |          |           |    |
|-----------------|----------|-----------|----|
| PLISGETEVCHA_R6 | 00000027 | RG        | 02 |
| SIZ...          | =        | 00000001  |    |
| STK_L_AP        |          | 00000008  |    |
| STK_L_ARG_LIST  |          | FFFFFFFF8 |    |
| STK_L_CND_HND   |          | 00000000  |    |
| STK_L_CND_LST   |          | FFFFFFFF4 |    |
| STK_L_DISPLAY   |          | FFFFFFFFC |    |
| STK_L_FP        |          | 0000000C  |    |
| STK_L_PC        |          | 00000010  |    |
| STK_L_PSL       |          | 00000004  |    |
| STK_L_REGS      |          | 00000014  |    |
| STR_B_FIELD     |          | 00000018  |    |
| STR_C_LEN       |          | 00000C08  |    |
| STR_L_FLD_END   |          | 00000014  |    |
| STR_L_FLD_PT    |          | 00000010  |    |
| STR_L_FP        |          | 00000004  |    |
| STR_L_FS        |          | 0000000C  |    |
| STR_L_PARENT    |          | 00000008  |    |
| STR_L_SP        |          | 00000000  |    |
| STR_L_STACK     |          | 00000C04  |    |
| STR_L_STACK_END |          | 00000408  |    |

PL  
1-

+-----+  
! Psect synopsis !  
+-----+

| PSECT name | Allocation       | PSECT No. | Attributes                                              |
|------------|------------------|-----------|---------------------------------------------------------|
| ABS        | 00000000 ( 0.)   | 00 ( 0.)  | NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE |
| \$ABSS     | FFFFFFFFC ( 0.)  | 01 ( 1.)  | NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE       |
| _PLISCODE  | 00000187 ( 391.) | 02 ( 2.)  | PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG         |

+-----+  
! Performance indicators !  
+-----+

| Phase                  | Page faults | CPU Time    | Elapsed Time |
|------------------------|-------------|-------------|--------------|
| Initialization         | 9           | 00:00:00.05 | 00:00:00.35  |
| Command processing     | 73          | 00:00:00.57 | 00:00:01.75  |
| Pass 1                 | 188         | 00:00:06.92 | 00:00:16.09  |
| Symbol table sort      | 0           | 00:00:00.66 | 00:00:01.63  |
| Pass 2                 | 56          | 00:00:01.30 | 00:00:02.91  |
| Symbol table output    | 10          | 00:00:00.07 | 00:00:00.29  |
| Psect synopsis output  | 1           | 00:00:00.03 | 00:00:00.03  |
| Cross-reference output | 0           | 00:00:00.00 | 00:00:00.00  |
| Assembler run totals   | 337         | 00:00:09.60 | 00:00:23.06  |

The working set limit was 900 pages.

35926 bytes (71 pages) of virtual memory were used to buffer the intermediate code.

There were 30 pages of symbol table space allocated to hold 574 non-local and 3 local symbols.

305 source lines were read in Pass 1, producing 11 object records in Pass 2.

20 pages of virtual memory were used to define 18 macros.

+-----+  
! Macro library statistics !  
+-----+

| Macro library name                       | Macros defined |
|------------------------------------------|----------------|
| _\$255\$DUA28:[PLIRTL.OBJ]PLIRTMAC.MLB;1 | 7              |
| _\$255\$DUA28:[SYSLIB]STARLET.MLB;2      | 7              |
| TOTALS (all libraries)                   | 14             |

623 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=TRACEBACK/LIS=LISS:PLIGETEDI/OBJ=OBJ\$:PLIGETEDI MSRC\$:PLIGETEDI/UPDATE=(ENH\$:PLIGETEDI)+LIB\$:PLIRTM



0308 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

PLIFORMAT  
LIS

PLIGETBUF  
LIS

PLMSGTXT  
LIS

PLIGETDI  
LIS

PLIHEEP  
LIS

PLIPUTFIL  
LIS

PLIRMSBIS  
LIS

PLIRECOPT  
LIS

PLIOPEN  
LIS

PLIREAD  
LIS

PLIPROTEC  
LIS

PLIREWRT  
LIS

PLIGETLIS  
LIS

PLIPUTDI  
LIS

PLIPKDIU  
LIS

PLIPUTLIS  
LIS

PLMSGPTR  
LIS

PLIPKDIU  
LIS

PLIPUTBUF  
LIS

PLIGETFIL  
LIS